

DETAILED ACTION

1. Claims 2-3, 15-16 and 28 have been canceled.
2. Claims 1, 4-14, 17-27, 29-31 and 43-48 are being allowed.

EXAMINER'S AMENDMENT

3. An examiner's amendment to the record appear below. Should the change and/or additions be unacceptable to the Applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such amendment, it MUST be submitted no later than the payment of issue fee.

Authorization for examiner's amendment was given in a telephone interview with Mark W. Raymond R. Tabandeh, Registration No. 43,945 on June 17, 2009 to obviate a potential 101 issues and put the case in condition for allowance.

4. The Claims are amended, as presented below, to adopt the changes indicated by Applicant's representative on June 17, 2009.

IN THE CLAIMS:

The claim listings below replace all prior versions, and listings, of claims in the application.

Please cancel claims 2-3, 15-16 and 28 and amend claims 1, 4-5, 14, 17-18, 27, 29-30, 43 and 45 as follows:

1. (Currently Amended) A method for automatically preventing errors in computer software having a plurality of different life cycle phases, the method comprising:

storing source code of the computer software in a code repository;

providing a plurality of software verification programs to verify the computer software, wherein each of the plurality of software verification programs relates to a respective lifecycle phase of the computer software, and automatically generates one or more test cases from the source code of the computer software, and wherein each of the plurality of software verification programs is an independent verification tool including a user interface and is capable of being executed in a batch process;

executing a first software verification program relating to a first lifecycle phase of the computer software, wherein the first software verification program automatically generates one or more test cases from the source code of the computer software;

executing a second software verification program relating to a next lifecycle phase of the computer software different from the first lifecycle phase, wherein the second software verification program automatically generates one or more test cases from the source code of the computer software;

generating verification results for the first and the next lifecycle phase of the computer software, responsive to executing the first and second software verification programs and the automatically generated test cases;

processing the verification results for generating a representation of functional behavior of the computer software[[.]];

providing a common configuration file for the plurality of verification programs;

customizing a verification scope of one or more of the verification programs by modifying the common configuration file responsive to an objective criterion of quality of the computer software wherein the objective criterion of quality is a quality rating of the entire computer software that takes into account the verification results of each of the verification programs, the number of test cases executed, and the success or failure of the test cases.

2. (Canceled)

3. (Canceled)

4. (Currently Amended) The method of claim [[3]] 1 further comprising modifying a portion of the common configuration file specific to one of the plurality of verification programs responsive to the objective criterion of quality of the computer software.

5. (Currently Amended) The method of claim [[3]] 1 further comprising modifying a portion of the common configuration file specific to one of a plurality of software developers responsive to the objective criterion of quality of the computer software.

6. (Previously Presented) The method of claim 1, further comprising formulating the verification results in a confidence factor represented by the equation:

$$C = p/t \times 100,$$

where p is number of successful test cases and t is total number of test cases.

7. (Original) The method of claim 1, wherein each portion of the computer software being developed by a software developer of a plurality of software developers, and the verification results include a plurality of objective criteria each of the plurality of objective criteria corresponding to quality of a respective portion of the computer software developed by each software developer of the plurality of software developers.

8. (Previously Presented) The method of claim 7 further comprising providing a common configuration file for the plurality of verification programs; and modifying the common configuration file responsive to one or more objective criteria corresponding to quality of a respective portion of the computer software developed by each software developer.

9. (Previously Presented) The method of claim 7 further comprising verifying a first portion of the computer software developed by a first developer of the plurality of software developers using the plurality of verification programs, before the computer software is stored in the code repository.

10. (Original) The method of claim 9 further comprising allowing storing the first portion of the computer software in the code repository only if result of verification of the first portion meets a set standard.

11. (Original) The method of claim 10 further comprising modifying the set standard responsive to the objective criterion of quality of the computer software.

12. (Original) The method of claim 10, wherein the set standard is common to each of the plurality of software developers.

13. (Original) The method of claim 10, wherein the set standard is unique to at least one of the plurality of software developers.

14. (Currently Amended) A system for automatically preventing errors in computer software having a plurality of different life cycle phases comprising:

a processor for executing computer software;

a code repository for storing source code of the computer software;

means for providing a plurality of software verification programs to verify the computer software, wherein each of the plurality of software verification programs relates to a respective lifecycle phase of the computer software and automatically generates one or more test cases from the source code of the computer software, and wherein each of the plurality of software verification programs is an independent verification tool including a user interface and is capable of being executed in a batch process;

means for executing a first software verification program relating to a first lifecycle phase of the computer software, wherein the first software verification program automatically generates one or more test cases from the source code of the computer software;

means for executing a second software verification program relating to a next lifecycle phase of the computer software different from the first lifecycle phase, wherein the second software verification program automatically generates one or more test cases from the source code of the computer software;

means for generating verification results for the first and the next lifecycle phases of the computer software, responsive to executing the first and second software verification programs and the automatically generated test cases;

means for processing the verification results for generating a representation of functional behavior of the computer software[[.]];

means for providing a common configuration file for the plurality of verification programs;

means for customizing a verification scope of one or more of the verification programs by modifying the common configuration file responsive to an objective criterion of quality of the computer software wherein the objective criterion of quality is a quality rating of the entire computer software that takes into account the verification results of each of the verification programs, the number of test cases executed, and the success or failure of the test cases.

15. (Canceled)

16. (Canceled)

17. (Currently Amended) The system of claim [[15]] 14 further comprising means for modifying a portion of the common configuration file specific to one of the plurality of verification programs responsive to an objective criterion of quality of the computer software.

18. (Currently Amended) The system of claim [[15]] 14 further comprising means for modifying a portion of the common configuration file specific to one of a plurality of software developers responsive to an objective criterion of quality of the computer software.

19. (Previously Presented) The system of claim 14, further comprising means for formulating the verification results in a confidence factor represented by the equation:

$$C = p/t \times 100,$$

where p is number of successful test cases and t is total number of test cases.

20. (Original) The system of claim 14, wherein each portion of the computer software being developed by a software developer of a plurality of software developers, and the

verification results include a plurality of objective criteria each of the plurality of objective criteria corresponding to quality of a respective portion of the computer software developed by each software developer of the plurality of software developers.

21. (Previously Presented) The system of claim 20 further comprising means for providing a common configuration file for the plurality of verification programs; and means for modifying the common configuration file responsive to one or more objective criteria corresponding to quality of a respective portion of the computer software developed by each software developer.

22. (Previously Presented) The system of claim 20 further comprising means for verifying a first portion of the computer software developed by a first developer of the plurality of software developers using the plurality of verification programs, before the computer software is stored in the code repository.

23. (Original) The system of claim 22 further comprising means for allowing storing the first portion of the computer software in the code repository only if result of verification of the first portion meets a set standard.

24. (Original) The system of claim 23 further comprising means for modifying the set standard responsive to the objective criterion of quality of the computer software.

25. (Original) The system of claim 23, wherein the set standard is common to each of the plurality of software developers.

26. (Original) The system of claim 23, wherein the set standard is unique to at least one of the plurality of software developers.

27. (Currently Amended) A method for automatically preventing errors in computer software having a plurality of different life cycle phases, the method comprising:

detecting an error in the computer software, the detected error belonging to a class of errors;

providing a plurality of software verification programs each of the plurality of software verification programs related to a respective lifecycle phase of the computer software, wherein each of the plurality of software verification programs is an independent verification tool including a user interface and is capable of being executed in a batch process;

analyzing the detected error in the computer software;

determining what phase of the lifecycle the detected error was introduced, based on analyzing the detected error;

providing a common configuration file for the plurality of verification programs;

customizing a verification scope of one or more of the plurality of verification programs that correspond to the determined lifecycle phase where the detected error was introduced;

executing the plurality of software verification programs to verify the class of the detected error is detected in a same lifecycle phase of the computer software as the determined lifecycle phase where the detected error was introduced[.],wherein the plurality of software verification programs automatically generate one or more test cases from a source code of the computer software;

generating verification results for the each lifecycle phase of the computer software, responsive to executing the plurality of software verification programs and the automatically generated test cases;

customizing a verification scope of one or more of the verification programs by modifying the common configuration file responsive to an objective criterion of quality of the computer software wherein the objective criterion of quality is a quality rating of the entire computer software that takes into account the verification results of each of the verification programs, the number of test cases executed, and the success or failure of the test cases.

28. (Canceled)

29. (Currently Amended) The method of claim [[28]] 27 further comprising modifying a portion of the configuration file specific to one of the plurality of verification programs based on the objective criterion of quality of the computer software.

30. (Currently Amended) The method of claim [[28]] 27 further comprising modifying a portion of the common configuration file specific to one of a plurality of software developers responsive to the objective criterion of quality of the computer software.

31. (Previously Presented) The method of claim 27, further comprising processing the verification results for generating an objective criterion of quality of the computer software by formulating the verification results in a confidence factor represented by the equation:

$$C = p/t \times 100,$$

where p is number of successful test cases and t is total number of test cases.

32. - 42 (Canceled)

43. (Previously Presented) The method of claim 28 further comprising customizing the verification scope of one or more of the plurality of verification programs for a second time, if the known error is not detected by executing the plurality of software verification programs.

44. (Previously Presented) The method of claim 27 further comprising executing the plurality of software verification programs periodically to prevent the known error from re-occurring when the computer software is modified.

45. (Currently Amended) A system for automatically preventing errors in computer software having a plurality of different life cycle phases comprising:

a processor for executing computer software;

means for detecting an error in the computer software, the detected error belonging to a class of errors;

means for providing a plurality of software verification programs each of the plurality of software verification programs related to a respective lifecycle phase of the computer software, wherein each of the plurality of software verification programs is an independent verification tool including a user interface and is capable of being executed in a batch process;

means for analyzing the detected error in the computer software;

means for determining what phase of the lifecycle the detected error was introduced, based on analyzing the detected error;

means for providing a common configuration file for the plurality of verification programs;

means for customizing a verification scope of one or more of the plurality of verification programs that correspond to the determined lifecycle phase wherein the detected error was introduced; and

means for executing the plurality of software verification programs to verify the class of the detected error is detected in a same lifecycle phase of the computer software as the determined lifecycle phase wherein the detected error was introduced[[.]],wherein the plurality of software verification programs automatically generate one or more test cases from a source code of the computer software;

means for generating verification results for the each lifecycle phase of the computer software, responsive to executing the plurality of software verification programs and the automatically generated test cases;

means for customizing a verification scope of one or more of the verification programs by modifying the common configuration file responsive to an objective criterion of quality of the computer software wherein the objective criterion of quality is a quality rating of the entire computer software that takes into account the verification results of each of the verification programs, the number of test cases executed, and the success or failure of the test cases.

46. (Previously Presented) The system of claim 45 further comprising means for executing the plurality of software verification programs to verify the known error is detected in computer software.

47. (Previously Presented) The system of claim 46 further comprising means for customizing the verification scope of one or more of the plurality of verification programs for a second time, if the known error is not detected by executing the plurality of software verification programs.

Art Unit: 2192

48. (Previously Presented) The system of claim 45 further comprising means for executing the plurality of software verification programs periodically to prevent the known error from re-occurring when the computer software is modified.

--End--

Allowable Subject Matter

5. The following is an examiner's statement of reasons for allowance:

As applicant pointed out under Remark section, pages 18-22, Jorapur (US 7,299,382 B2), taken either singly and/or in combination with other cited prior arts, do not teach the combined functional limitations of executing a first software verification program relating to a first lifecycle phase of the computer software, wherein the first software verification program automatically generates one or more test cases from the source code of the computer software; executing a second software verification program relating to a next lifecycle phase of the computer software different from the first lifecycle phase, wherein the second software verification program automatically generates one or more test cases from the source code of the computer software; generating verification results for the first and the next lifecycle phase of the computer software, responsive to executing the first and second software verification programs and the automatically generated test cases; processing the verification results for generating a representation of functional behavior of the computer software; providing a common configuration file for the plurality of verification programs; customizing a verification scope of one or more of the verification programs by modifying the common configuration file responsive to an objective criterion of quality of the computer software wherein the objective criterion of quality is a quality rating of the entire computer software that takes into account the verification results of each of the verification programs, the number of test cases executed, and the success or failure of the test cases, as recited in such manners in each of independent claims 1, 14, 27 and 45.

Prior arts of record do not teach and/or suggest these claimed limitations, thus, all remaining pending claims 1, 4-14, 17-27, 29-31 and 43-48 are allowed.

Any comments considered necessary by applicant must be submitted no later than the payment of the issue fee and, to avoid processing delays, should preferably accompany the issue fee. Such submissions should be clearly labeled "Comments on Statement of Reasons for Allowance."

Conclusion

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to ISAAC T. TECKLU whose telephone number is (571) 272-7957. The examiner can normally be reached on M-TH 9:300A - 8:00P.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q. Dam can be reached on (571) 272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Isaac T Tecklu/
Examiner, Art Unit 2192

/Tuan Q. Dam/
Supervisory Patent Examiner, Art Unit 2192